

# 밴드 인페인팅

## Band In-painting

이수빈 Subin Lee, 서용덕 Yongduck Seo  
서강대학교 미디어공학과

**요약** 본 논문은 이미지에서 불필요한 영역을 삭제하고, 그 영역을 배경과 어울리게 채워넣는 이미지 인페인팅 방법을 제안한다. 제안하는 인페인팅 방법은 크게 인페인팅 영역을 채우는 밴드 인페인팅(band in-painting)과 seamless cloning으로 나눌 수 있다. 밴드 인페인팅(band in-painting)은 인페인팅 영역의 경계를 따라서 일정한 두께를 가지는 타겟 밴드(target band)를 정의하고, 인페인팅 영역 밖의 모든 픽셀을 중심으로 하는, 타겟 밴드와 같은 모양과 크기를 가지는 소스 밴드(source band)와 타겟 밴드 차이를 계산하여, 그 값의 차이가 가장 작은 소스 밴드 영역의 값을 인페인팅 영역에 복사하는 것이다. Seamless cloning은 인페인팅 영역과 입력 이미지의 경계를 없애는 것이다.

**핵심어:** 이미지 인페인팅(Image inpainting), seamless cloning

### 1. 서론

최근 영화들 중에 특수효과가 사용되지 않은 영화가 없을 정도로 특수효과는 영화에서 광범위하게 사용되고 있다. 많은 특수 효과 기술들 중 디지털 인페인팅(Digital In-painting)은 실사 이미지에서 제거된 특정 영역을 주변 배경과 어울리게 채워 넣는 기술로 다양한 부분에서 이 기술이 사용되고 있다. 영화 촬영 후 영화 속 배경이나 시대에 맞지 않은 물체나 사람이 있을 경우 영화를 재촬영할 수도 있지만, 인페인팅 기술을 사용하면 쉽게 물체나 사람을 지울 수 있어 시간과 비용을 절감할 수 있다. 또한 훼손된 필름이나 비디오 영상자료를 복원하기 위하여 인페인팅 기술을 사용하면 스크래치나 노이즈 등 좁은 영역을 채울 수 있다. 본 연구는 디지털 인페인팅 기술에 대한 것으로, 먼저 관련 연구에 대해 설명하고, 다음으로 본 논문에서 제안하는 방법과 실험 결과를 보이고자 한다.

### 2. 관련 연구

이미지 인페인팅 방법은 크게 두 가지로 나누어 볼 수 있다. 첫 번째는 작은 영역을 채우는 방법에 대한 연구로 [1, 2]가 있다. 이 방법은 PDE를 이용하여 보간(interpolation)

하는 것으로, smooth continuation을 나타내는 이미지 영역에 대해서 효과적이다. 그림 1은 PDE를 이용하여 보간(interpolation)으로 작은 영역(붉은색 글자들)을 인페인팅한 결과로 인페인팅 영역 주변 색상이 유사하면 좋은 결과를 보이지만, 텍스처가 있는(textured) 영역에서는 주변 색상이 서로 다르기 때문에 blurring이 발생하여 좋지 않은 결과를 보인다.

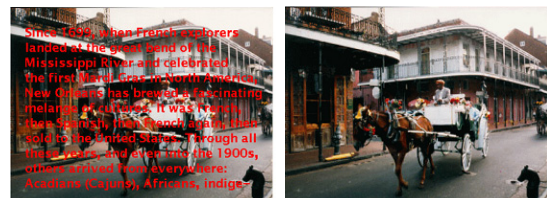


그림 1. (Small Target Area: [1]의 결과)

두 번째는 큰 영역을 채우는 방법에 대한 연구로 [3, 5, 7]이 있다. 이 방법은 텍스처 합성(texture synthesis)을 이용하여 인페인팅 영역을 채워주는 것으로, structural continuity를 가지는 이미지에 대해서는 좋은 결과를 보인다. 그림 2는 텍스처 합성을 이용하여 큰 영역(흰색 부분)을

인페인팅 한 결과로 주변과 연속성을 가지는 인페인팅 영역에 대해 좋은 결과를 보인다. 하지만 인페인팅 텍스처가 없고(non-textured) 부드러운(smooth) 영역에서는 structural continuity에 대한 특징을 정할 수 없기 때문에 효과적이지 않다.



그림 2. (Large Target Area: [3]의 결과)

본 연구는 큰 영역을 인페인팅하는 방법을 제안한다. [3]과 마찬가지로 텍스처 영역을 이용하여 인페인팅 영역을 채우지만, [3]에서는 인페인팅 영역을 여러 개의 패치(patch)로 나눠 각 각을 채워 주는 반면, 제안하는 방법은 인페인팅 영역을 하나의 패치로 생각하여 한 번에 채운다. 패치를 여러 개로 나누어 채우면 패치들마다 계산이 필요하기 때문에 시간이 오래 걸리고, 인페인팅 영역 내에서 패치들 사이에 연속성이 없을 경우 좋지 않은 결과가 나타날 수 있다. 하지만 제안하는 방법(band in-painting)은 하나의 패치만 계산하기 때문에 이전 방법보다 속도가 더 빠르고, 하나의 패치로 인페인팅 영역을 채우기 때문에 해당 영역 내에서는 자연스러운 결과가 나타난다. 표 1은 [3]과 제안하는 방법을 비교한 것이다.

표 1. ([3] 과 제안하는 방법의 비교)

	Exemplar-based Inpainting[3]	Band In-painting (제안하는 방법)
수행시간	약 4분	약 14초
조 건	structural continuity	인페인팅 영역 크기의 제한

표 1에서 수행시간은 인페인팅 영역이 전체 입력 이미지의 약 0.5%를 차지하는 그림 4 (a)를 가지고 인페인팅을 수행한 시간이다. 수행시간을 비교하면, 제안하는 방법이 exemplar-based in-painting[3]에 비해 더 빠르다는 것을 알 수 있다. 다음 장에서는 제안하는 방법과 실험 결과에 대해 자세히 설명한다.

### 3. 제안하는 인페인팅 방법

그림 3은 제안하는 인페인팅 방법에 대한 과정을 간단히 보여준다.

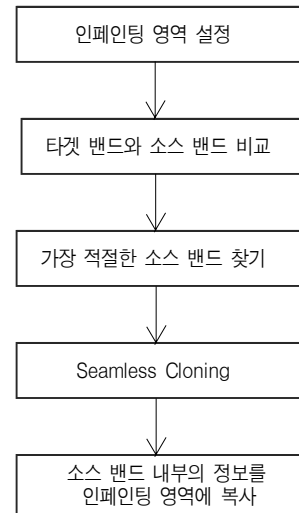


그림 3. (제안하는 인페인팅 과정)

먼저 지우고자 하는 영역, 즉 인페인팅 하고자 하는 영역을 사용자가 정해주고, 영역의 경계(boundary)를 따라서 일정 두께의 타겟 밴드(target band)를 정한다. 다음으로, 인페인팅 영역 밖의 모든 픽셀에 대해, 그 픽셀을 중심으로 하는, 타겟 밴드와 같은 모양과 크기를 가지는 소스 밴드(source band)와 타겟 밴드 차이를 계산하여, 그 값이 가장 작은 소스 밴드 내부의 영역을 인페인팅 영역(target band 내부)에 복사한다. 마지막으로 인페인팅 영역과 입력 이미지의 경계에 대한 seamless cloning을 계산하여 경계가 사라지도록 한다.

제안하는 인페인팅 방법은 크게 인페인팅 영역을 채우는 밴드 인페인팅(band in-painting)과 인페인팅 영역과 입력 이미지 사이의 경계를 없애는 seamless cloning으로 나눌 수 있다. 3.1에서 밴드 인페인팅에 대해 설명하고, 3.2에서 seamless cloning에 대해 설명한다.

#### 3.1 밴드 인페인팅 (Band In-painting)

밴드 인페인팅의 첫 번째로 단계로 사용자가 이미지에서 인페인팅 하고자 하는 영역을 수동으로 정한다. 그림 4 (a)에서 흰색 부분이 인페인팅을 하고자 하는 영역이고, 붉은색 선은 인페인팅 영역의 경계(boundary)를 나타낸다.

두 번째 단계로, 인페인팅 영역의 경계를 따라서 일정한 두께를 가지는 타겟 밴드를 결정한다. 타겟 밴드는 이미지 내의 각 각의 소스 밴드와 비교를 위해 사용된다. 여기서 소

스 밴드란 전체 이미지에서 인페인팅 영역을 제외한 부분을 타겟 밴드와 같은 모양으로 나눈 것을 말한다.

세 번째 단계로, 타겟 밴드와 각 소스 밴드 사이의 SSD(sum of squared difference)를 계산하여 그 값이 최소가 되는 소스 밴드를 찾는다. SSD의 계산의 다음과 같다.

$$SSD_k = \sum_{i=1}^N \left( (R_i^{sk} - R_i^t)^2 + (G_i^{sk} - G_i^t)^2 + (B_i^{sk} - B_i^t)^2 \right) \quad (1)$$

식 (1)에서 s는 소스 밴드, t는 타겟 밴드, R, G, B는 픽셀(pixel)의 각 색상 채널을 나타낸다. N은 밴드(band)의 총 픽셀 수, k는 각 각의 소스 밴드를 나타낸다.

마지막 단계로, 소스 밴드내부의 색상 값을 인페인팅 영역에 채워준다.

그림 4는 밴드 인페인팅 과정을 보여준다.

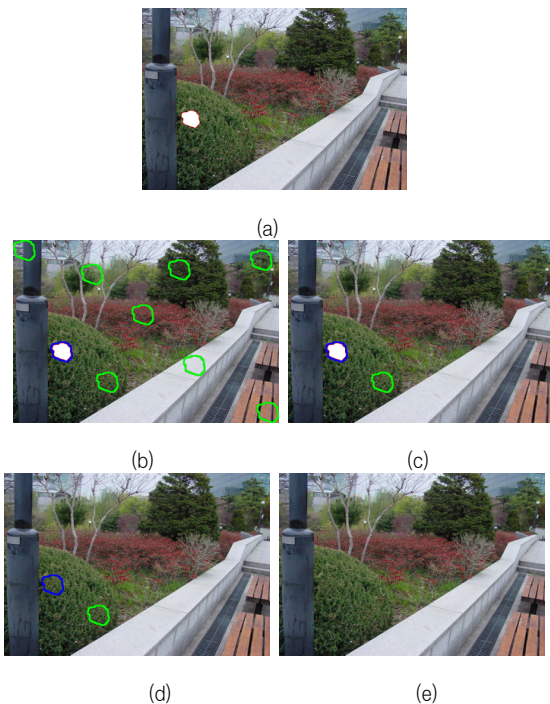


그림 4. (밴드 인페인팅 과정: (a) 인페인팅 영역 설정, (b) 타겟 밴드와 소스 밴드의 비교, 파란색 선-타겟 밴드, 녹색 선-소스 밴드들 (c) 타겟 밴드와 가장 유사한 소스 밴드의 위치 (d) 타겟 밴드 내부에 소스 밴드 내부 정보를 복사한 결과 (e) 최종 결과)

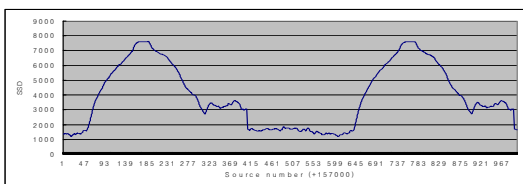


그림 5. (각 타겟 밴드와 소스밴드 사이의 SSD)

그림 4의 경우 총 소스 밴드의 개수가 204366개이고, 그 중에서 157606번째의 소스 밴드가 가장 작은 SSD를 가진다. 그림 5는 소스 밴드 157001번부터 157999번사이의 소스 밴드와 타겟 밴드 사이의 SSD를 계산하여 표로 나타낸 것이다.



그림 6. (결과 비교: (a) 밴드 인페인팅 결과, 계산 시간 약 14초 (b) [2]의 결과, 계산 시간 약 4분)

그림 6에서 (a)의 계산 시간은 약 14초, (b)의 계산 시간은 약 4분으로 제안하는 방법의 속도가 더 빠르다. 이유는 밴드 인페인팅 경우 소스 밴드들과 타겟 밴드를 한 번씩만 비교하면 되지만, exemplar-based in-painting의 경우 채우고자 하는 각 패치마다 소스 패치들과 반복하여 비교해야 하므로 인페인팅 영역을 모두 채우기 위한 시간이 더 오래 걸린다. 또한 exemplar-based in-painting의 결과는 인페인팅 영역 안이 여러 개의 패치들로 채워지기 때문에 패치들 사이에 연속성이 없을 경우 좋지 않은 결과를 보이는 반면, 밴드 인페인팅 결과는 하나의 패치를 붙이기 때문에 인페인팅 영역이 자연스럽게 채워진 것을 볼 수 있다.

그림 7은 밴드 인페인팅의 결과들이다. 그림 7에서 (a)열은 입력 이미지를 나타내고, (b)열은 밴드 인페인팅 결과를 나타낸다. 밴드 인페인팅은 주변의 색상이나 텍스처가 유사한 이미지에서 좋은 결과가 나타나는 것을 볼 수 있다.





그림 7. (밴드 인페인팅 결과: (a) 입력 이미지, (b) 밴드 인페인팅 결과)

### 3.2 Seamless Cloning

하나의 패치를 사용하여 인페인팅을 하면 인페인팅 영역은 자연스럽게 채워지지만, 인페인팅 영역과 입력 이미지 사이에는 경계(seam)가 나타나는 것을 볼 수 있다. 그림 8에서 (b)는 밴드인페인팅의 결과로, 인페인팅 영역과 입력 이미지 사이에 경계가 나타나는 예를 보여준다. 이러한 현상이 발생하는 이유는 그래디언트(gradient)의 차이가 크게 나타나기 때문인데, 이러한 현상을 해결하기 위해 Poisson cloning [6] 또는 covariant derivatives [4]을 이용하는 방법이 있다. 여기서서는 기존 covariant derivatives을 이용해 seamless cloning을 수행하여 경계가 나타나는 밴드 인페인팅 결과를 개선하고자 한다. Seamless cloning의 자세한 방법은 다음과 같다.



그림 8. (경계(seam)가 나타나는 예: (a) 입력 이미지, (b) 밴드 인페인팅 결과)

첫 번째 단계로, 인페인팅 영역의  $h(x,y)$ 는 초기 값을 1로 설정하고, 타겟 밴드 영역의  $h(x,y)$ 는 다음의 식을 사용하여 계산한다.

$$h(x,y) = \frac{f(x,y)}{g(x,y)} \quad (2)$$

$f(x,y)$ 는 타겟 밴드,  $g(x,y)$ 는 소스 밴드의 색상 값이다.

두 번째 단계로, 인페인팅 영역의  $h(x,y)$  값을 반복계산(iteration)을 통하여 업데이트(update)한다.

$$h^n(x,y) = K * h^{n-1}(x,y) \quad (3)$$

위의 식에서  $K$ 는  $(x,y)$ 를 중심으로 상하, 좌우는 1의 값을 가지고 나머지는 0의 값을 가지는 3x3의 마스크(mask)를 나타낸다.

0	1	0
1	0	1
0	1	0

그림 9. (K 마스크(mask))

마지막 단계로, 새로 계산된 인페인팅 영역의 색상 값은 마지막으로 계산된  $h(x,y)$ 의 값과 소스 밴드의 색상 값의 곱으로 계산한다.

$$\hat{f}(x,y) = h^n(x,y)g(x,y) \quad (4)$$

다음은 밴드 인페인팅 결과에 seamless cloning을 수행한 결과이다.

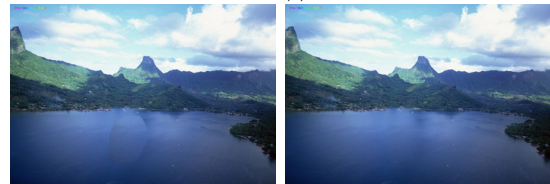


그림 10. (Seamless Cloning 결과: (a) 원본 이미지, (b) 밴드 인페인팅 결과, (c) Seamless Cloning 결과)



그림 10를 보면 밴드 인페인팅 결과는 인페인팅 영역과 원본 이미지 사이의 경계가 명확하게 보이지만, 인페인팅 영역에 대해 seamless cloning을 수행하면 인페인팅 영역과 입력 이미지 사이의 경계가 부드럽게 변한 것을 볼 수 있다. 그림 10의 경우 인페인팅 영역이 전체 이미지의 약 3%를 차지하고 밴드 인페인팅의 수행시간은 약 39초, seamless cloning 수행시간은 약 1초로 총 약 40초의 수행 시간이 걸린다.

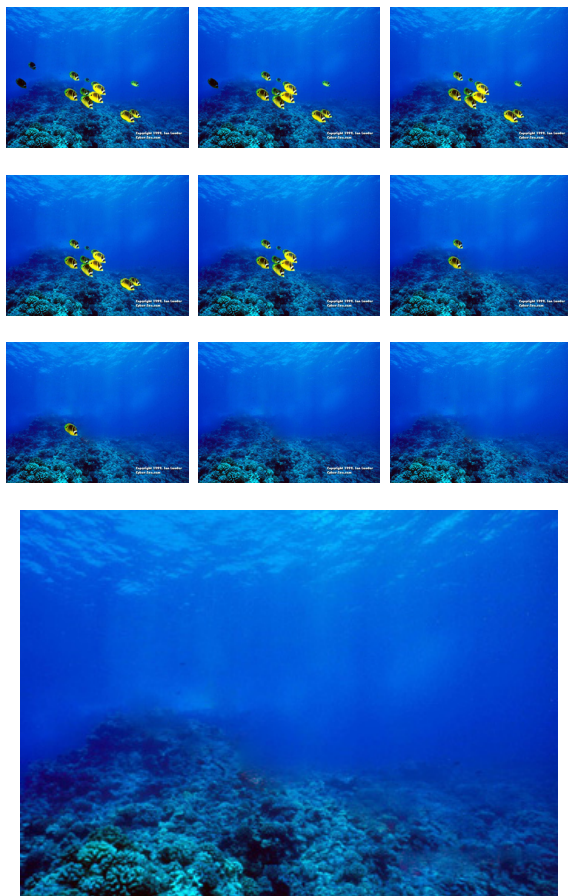


그림 11. (결과 이미지들)

그림 11은 하나의 이미지를 여러 개의 영역으로 나눠 나열된 순서대로 밴드 인페인팅을 수행하고 seamless cloning을 한 결과이다.

#### 4. 결론

제안하는 방법은 인페인팅 하고자 하는 영역을 둘러싸는 타겟 밴드를 구하고, 그 타겟 밴드와 모양이 같은 소스 밴드를 입력 이미지 내에서 모두 찾아 색상 차이를 비교하여, 그 차이가 가장 작은 소스 밴드 내부의 값을 인페인팅 영역에 채우는 것이다. 그리고 밴드 인페인팅의 결과로 인페인팅 영역과 입력 이미지 사이의 경계가 나타나는 부분은 seamless cloning을 수행함으로써 경계 부분을 부드럽게 바꾼다. 제안하는 방법은 기존 방법에 비해 빠르고, 좋은 결과를 보인다. 하지만 타겟 밴드의 크기가 커서 입력 이미지에서 같은 크기와 모양의 소스 밴드를 찾을 수 없는 경우에는 이 방법을 사용할 수 없고, 입력 이미지에서 불연속성이 존재하는 영역의 경우 좋지 않은 결과를 보이기도 한다.

#### Acknowledgment

본 연구는 문화관광부 및 한국문화콘텐츠진흥원의 문화콘텐츠기술연구소(CT)육성사업의 연구결과로 수행되었고, 서울시 신기술 연구개발 지원사업으로 수행되었다.

#### 참고문헌

- [1] Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, Coloma Ballester, "Image Inpainting", In Proc. ACM SIGGRAPH 2000, pp.417-424
- [2] Tony F. Chan, Jianhong Shen, "Mathematical Models for Local Nontexture Inpaintings", Society for Industrial and Applied Mathematics (SIAM) Journal on Applied Mathematics 2002, pp.1019-1043.
- [3] A. Criminisi, P. Perez, K. Toyama, "Object Removal By Exemplar-Based Inpainting", In Proc. Conference on Computer Vision and Pattern Recognition (CVPR) 2003, pp.721-728
- [4] Todor Georgiev, "Covariant Derivatives and Vision", In Proc. European Conference on Computer Vision (ECCV) 2006, pp.56-69
- [5] Nikos Lomodakis, Georgios Tziritas, "Image Completion Using Global Optimization", In Proc. Conference on Computer Vision and Pattern Recognition (CVPR) 2006, pp.442-452

[6] Patrick Perez, Michel Gangnet, Andrew Blake,  
"Poisson Image Editing", In Proc. ACM SIGGRAPH  
2003, pp.313–318

[7] Hitoshi Yamauchi, Jorg Gaber, Hans–Peter Seidel,

"Image Restoration using Multiresolution Texture  
Synthesis and Image Inpainting", In Proc.  
Computer Graphics International (CGI) 2003,  
pp.120–125